



WEB OF FRAIL DELIGHT

Robert Filman • Lockheed Martin • filman@ict.atc.lmco.com
Feniosky Peña-Mora • MIT • feniosky@mit.edu

As our friend Martin Yonke once observed, "Standards are wonderful. I think everyone should have one of their own." In that spirit, the Spider set out in search of the rules of the Web.

World Wide Web

Consortium • www.w3.org/
Our first stop, the World Wide Web Consortium (W3C), develops World Wide Web protocols and standards, and disseminates these standards through its Web site. The site includes links to publications such as the *World Wide Web Journal*, workshops and conferences, and mailing lists. Technology resident at the W3C includes WWW information, reference implementations, and sample applications.

The site ranges from the modest to the grandiose. A judicious 1992 vision statement asserts, "The WWW project merges the techniques of networked information and hypertext to make an easy but powerful global information system." But later on the same page we are told, "In fact, the Web is an architecture which will also embrace any future advances in technology, including new networks, protocols, object types, and data formats." We envy the clarity of their crystal ball.

Similarly, we would be more cautious than to proclaim the seven-year-old Web, "the embodiment of human knowledge." (As an embodiment of human knowledge, the Web mostly reminds the Spider of Borges's Library of Babel.)

Moving on to content, the W3C home page offers pointers both to technical information (divided into user interface; "technology and society," encompassing topics such as metadata, commerce, and digital signatures; architecture; and Web accessibility); and organizational data (news, conferences, history, people, and mailing lists). Our explorations were occasionally blocked by our lack of membership in the W3C. Like too much in life, one must pay for the better stuff.

We poked at "style sheets" and found the **Cascading Style Sheets**, level 2 initiative. (W3C uses "cascade" in the same way programmers use "inherit.") HTML has traditionally been a document-structuring language. Document rendering (for example, choice of fonts and font sizes) has been left to the browsing program. To give authors greater control over documentation, W3C introduced cascading style sheets. The level 2 initiative extends these ideas to include text vocalization (volume,

pauses, voice characteristic, speaker location); greater control on text positioning (layers, boxes within paragraphs, and similar layout tricks); differing styles for different print media; and better font specification, including the ability to download fonts.

It took considerably more poking at the HTTP-NG (next generation) material to find out what was going on. There are many pointers to documents that can best be described as "of historical interest." The most concise summary we found was at <http://www.w3.org/TR/1998/WD-HTTP-NG-goals>, where the project goal was stated as testing "the hypothesis that the current HTTP/1.X approach to Web protocol design can be replaced with one in which the Web is expressed as a particular set of interfaces on top of a generic distributed object system designed with Internet constraints in mind." The document continues, "In particular, we would like the generic distributed object system to be simple, yet rich enough to meet the semantic and performance requirements of CORBA, DCOM, and Java RMI. This doesn't mean we intend to unify the object models of CORBA, DCOM, and Java RMI (which are somewhat diverse and warty). It means that we'd like to move HTTP onto a generic distributed object system of analogous strength, so that programmers (and middleware-owning organizations) willing to switch could get roughly the same jobs done (at least)." That is, HTTP 1.1 removed the worst inefficiencies of HTTP: connections are kept and a variety of data can be transmitted over a single connection. In HTTP-NG, W3C has designs on the territory now occupied by middleware like CORBA, Java, and distributed COM.

The metadata activity seeks to formalize and generalize the ability to attach descriptive information to Web documents. The focus of this activity is the Resource Description Framework (RDF), which provides a semantic-net-like mechanism for describing relationships.

These are only a few of the W3C initiatives. Always the scholar, the

Spider was pleased to find a reference to a bibliography of Web information. However, when we followed the link to <http://www.w3.org/Bibliography.html>, we were warned that the document was three years out of date and directed to a better bibliography, which itself 404'd—a lucid illustration of the Web as a comprehensive embodiment of human knowledge.

Overall, there's a lot here, though the Spider expected it to be better organized. W3C is a standards organization, and as standards organizations are wont to do, there's more trees here than forest.



Object Management Group

• www.omg.org

A fundamental dualism in computer science is the division between data and process. The World Wide Web is a data-centric model for distributed computing. Web pages present information with a single data-focused operation ("Follow that link"). (Yes, we're oversimplifying.) The most intellectually coherent, process-centered system for distributed computing is the Object Management Group's Common Object Request Broker Architecture. CORBA describes the things (objects) that exist and the operations that can be performed by these objects. Programs can invoke operations; objects can choose how to implement particular operations.

The major virtue of CORBA is interoperability: programs running on different operating systems, developed in different languages, can effectively work together. This is accomplished by providing a common language for describing component interfaces (the Interface Definition Language) and standardizing the mapping (binding) between terms in IDL and specific programming languages such as C++, Smalltalk, Ada, and Java. (We heard it announced recently that OMG now has a binding to MUMPS. OMG will soon be working on bindings to other childhood diseases.) The major disadvantage of CORBA is that not being native to any particular environment, it requires programmer accommodation in every environment. OMG has branched out beyond providing standards for the architecture of distributed

systems to standards for vertical application domains, though with less demonstrable success. (Can you spell "standards without implementations"?)

The Spider's first impression of the OMG Web site was that he had landed not in a standards organization but a business, and a rather embattled business at that. "About OMG" starts off with "some general information about our company," and progresses to pages about organization history, people, and corporate memberships. Corporate success stories with CORBA are prominently featured, though the argument seems to be based more on breadth than depth. A site map and the technical library both allow for an icy plunge into the cryptic depths of committee discussions. The site lacks a search function—a surprising omission, given the diversity of material, the variety of users, and the presumed sophistication of the sponsoring organization.

CORBA is a complex architecture, centered on the invocation of requests

on objects, surrounded by a number of services (such as secure transactions, event notification, and naming), and with tentacles reaching outward to domains as diverse as insurance and life sciences research. It's easy to get lost, and the site map, while pretty, is not useful enough. The Spider was relieved when he finally stumbled across the "beginner's guide." This page features explanations at a number of different levels, including the OMG's brief "What is CORBA," Kate Keahey's CORBA tutorial, and Alan Pope's "CORBA." This last document provides a fair amount of detail about the CORBA implementation mechanism and touches on (and provides vital pointers to) the more ancillary activities.



The news section of the OMG site devotes two articles to refuting an OVUM report on the imminent death of CORBA. While OMG does a good job of trashing that particular

The Spider's rating system:



5 Flies = Scrumptious



4 Flies = Tasty



3 Flies = Filling



2 Flies = Edible



1 Fly = Yuck!

URLs for this column

About OMG • www.omg.org/about/omaov.htm

Alan Pope's "CORBA" • www.qds.com/people/apope/Corba/index.html

Beginner's guide • www.omg.org/news/begin.htm

CSS • www.w3.org/Style/Activity.html

Kate Keahey's CORBA tutorial • www.cs.indiana.edu/hyplan/kksiazek/tuto.html

Library of Babel • jubal.westnet.com/hyperdiscordia/library_of_babel.html

Metadata activity • www.w3.org/Metadata/Activity.html

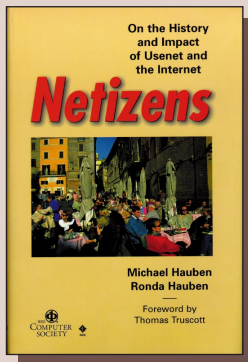
Progress on HTTP-NG • www.w3.org/Protocols/HTTP-NG/http-ng-status.html

Protocol extension protocol • www.w3.org/Protocols/PEP/

What is CORBA • www.omg.org/about/wicorba.htm

World Wide Web Journal • www.w3j.com/

Go to the
Online Bookstore
<http://computer.org>



A fascinating account of the past, present, and future of the Internet

Netizens
On the History and
Impact of Usenet
and the Internet
by Michael Hauben
and Ronda Hauben

Netizens looks at the creation and development of the global computer network. The book is a detailed description of the Net's construction and a step-by-step view of the past, present, and future of Usenet and the Internet.

384 pages. 6" x 9" Hardcover.
May 1997. ISBN 0-8186-7706-6.
Catalog # BP07706
\$25.00 Members / \$28.95 List

Call toll-free:
+1 800.CS.BOOKS



10662 Los Vaqueros Circle
Los Alamitos, CA 90720-1314

report, the articles (and the site) leave the impression, methinks, that the lady doth protest too much.

In reality, CORBA is a clever solution to a particular problem—getting systems written in a variety of languages on a variety of platforms to interoperate efficiently. But CORBA is not easy to use. OMG's difficulty is that for less complex problems, simpler solutions are appealing: HTTP, which offers less efficient but easy interoperable, distributed information access (and rapidly improving efficiency and expressiveness); Java, which offers straightforward distributed programming if only you're willing to program in Java; and Microsoft's DCOM, which provides straightforward multilingual computation at the minor costs of restricting your programming to the 95 percent of the world's computers that run Windows and indenturing your first-born child to Bill Gates.

However, any conceptual difficulty inherent in CORBA pales before DCOM, whose expression is intimately wrapped up with its environment. For a comparison of the two, we turned to Tallman and Kain.

COM versus CORBA: A Decision Frame • www.quininc.com/quininc/COMCORBA.html

Owen Tallman and J. Bradford Kain COM is a Microsoft product that was developed as an integration scheme to support compound documents. Originally designed for a single address space, it has grown like Topsy to encompass distributed processing. COM is a binary integration scheme—its implementation is centered on dynamically linking libraries through tables of function pointers. This works quite efficiently for C++ implementations on Wintel machines (as C++ is based on pointer tables, and Wintel binaries can be readily swapped) but less well in other languages and not at all on other platforms. Being a binary scheme for a single platform allows the compiler to hide more of the distributed aspects of programming than CORBA, which produces source-level files that then require compilation and linking with an application.

Microsoft controls the specification of COM and does not seem reluctant to tailor the definition to efficient interaction with Windows. This has the disadvantage of discouraging free-market innovation, but the advantage of providing an authoritative implementation that works well in its target market. Since that market may seem ubiquitous, many software developers are content under the COM tent.

The authors compare the characteristics of the two systems.

- **Object models.** CORBA has objects that support interfaces; COM's more complex system is based on class objects.
- **Errors and exceptions.** CORBA has a standard exception model; COM returns error values.
- **Object identity and persistence.** CORBA has a stronger notion of object identity and more straightforward paths to object persistence.
- **Scalability.** Neither CORBA nor COM can make a convincing case for scalability, but at least the CORBA architecture's support for heterogeneity would seem to give it an advantage.
- **Services.** CORBA has many well-defined, precisely architected, though not necessarily implemented services, while a COM programmer can take nimble advantage of everything in the Windows operating system.
- **Platform support.** CORBA runs on a large variety of platforms; COM is well integrated with one.
- **Maturity.** CORBA has many more deployed applications. OMG has been more energetic about showcasing its successes than Microsoft.

The Spider likes this paper, despite coming away with the feeling that CORBA is winning every battle but losing the war. 🕷️ 🕷️ 🕷️ 🕷️ 🕷️

We've gone on long enough. The other major network distributed computing model, Java, deserves discussion, but there's enough there to fill a (future) column. And our thanks to Anna Hemstead Branch for the column title. ■